



Proposta: Sistema de gerenciamento de empregados da empresa IMDCorp.

Funcionalidades e características:

O sistema a ser desenvolvido deve ser capaz de criar e gerenciar empregados da empresa IMDCorp

Classes para criação:

Classe Endereco

Crie uma classe chamada “Endereço” com os atributos privados:

- string rua;
- int numero;
- string bairro;
- string cidade;
- string cep;

Dentro da classe “Endereco” crie todos os métodos Get/Sets públicos para os atributos.

Classe Pessoa

Crie uma classe chamada “Pessoa” com os seguintes atributos privados:

- string nome;
- string cpf;
- string dataNascimento;
- string genero;
- Endereco endereco;

Dentro da classe “Pessoa” crie um construtor parametrizado e todos os métodos Get/Sets públicos para os atributos..

Classe Funcionário

Crie uma classe chamada “Funcionario” com os seguintes atributos privados:

- string matricula;
- float salario;
- string departamento;
- int cargaHoraria;
- string dataIngresso;

Dentro da classe crie um construtor parametrizado e todos os métodos Get/Sets dos atributos dessa classe (públicos).

Classe Professor

Dentro desta classe crie duas enumerações:

- enum Nivel: (I, II, III, IV, V, VI, VII, VIII)
- enum Formacao: (ESPECIALIZACAO, MESTRADO, DOUTORADO)

e dois atributos privados dessas enumerações:

- Nivel nivelProfessor;
- Formacao formacaoProfessor;
- string disciplina;

Crie um construtor parametrizados para inicializar todos os atributos da classe e os métodos Get/Set (públicos).

Classe TecnicoADM

Dentro desta classe crie os atributos privados :

- float adicional produtividade= 0.25;
- string funcaoDesempenhada;

um construtor parametrizado e os métodos Get/Set necessários (públicos).

Classe BancoDAO

Crie uma classe chamada “BancoDAO” que armazene e gerencie um vector do tipo “Professor” e um vector do tipo “TecnicoADM”:

- vector<Professor> professores;
- vector<TecnicoADM> tecnicosADM;

Dentro dessa classe crie os seguintes métodos públicos:

- cadastrarProfessor() - Cadastra um novo professor

- cadastrarTecnicoADM() - Cadastra um novo tecnicoADM
- listarProfessores() - Lista todos os professores por nome e disciplina;
- listarTecnicosADM() - Lista todos os tecnicosADM por nome e função;
- deletarProfessor(int matricula) - Apaga um professor a partir de sua matrícula
- deletarTecnicoADM(int matricula) - Apaga um TecnicoADM a partir de sua matrícula
- buscarProfessor(int matricula) - Localiza e exibe um professor a partir de uma matrícula passada como parâmetro.
- buscarTecnicoADM(int matricula) - Localiza e exibe um TecnicoADMa partir de uma matrícula passada como parâmetro.

Informações gerais

Essas informações deverão ser salvas em um arquivo (funcionarios.txt) ao fim do programa e lidas sempre que o programa iniciar para que o sistema já tenha o que exibir.

Crie no mínimo 5 professores e 5 tecnicosADM (10 funcionários).

Todas as classes deverão estar modularizadas (arquivos cpp e hpp).

crie um menu no qual é possível escolher qual função chamar. a

Critérios de avaliação

- Código corretamente modularizado (2,0);
- Criação das classes solicitadas (1,0);
- Informações salvas nos arquivos (2,0);
- Utilização da herança múltipla (2,0);
- Implementação correta de todos os métodos solicitados (2,0);
- Código organizado e comentado (1,0)

Entregáveis

O código deverá ser zipado e enviado via sigaa (tarefa). Envie na pasta zipada um arquivo (readme.txt) explicando como compilar e executar as funções (a ausência desse arquivo implicará 1,0 ponto de desconto na nota).